Compilation – TD3 : Données Structurées

Nawfal ‘Massine’ MALKI – 4991 – STI 3A TD3

Exercice 1 : Machine à registres

1. Où se trouvent x et y ? Donnez ρ(x) et ρ(y). Comment le résultat est-il retourné ?

f :

@x = bp+3 @y = bp+2
ρ(x) = t1 ρ(y) = t2

g :

@x = bp+3 @y = bp+2
ρ(x) = t1 ρ(y) = t2

2. Retrouvez la fonction en C

f :

```
int f(int x, int y) {
     return x+y-1;
}
```

g :

```
int g(int x, int y) {
     int result;
     if (x<y)
          result = f(x,x-y);
     return result;
}
```

3. Donner une réalisation possible des temporaires t1, t2, … avec des registres x86-64 parmis $rax, $rbx, $rcx, $rdx, $r8, …, $r15 ($bp est réalisé par $rbp).

Le registre contenant la valeur de retour doit être rax.

f : le temporaire de retour est t6.

t	t1	t2	t3	t4	t5	t6
Registre	Rbx	Rcx	Rdx	R8	R9	rax

g : deux possibilités : soit t7 soit t4

t	t1	t2	t3	t4	t5	t7
Registre	Rbx	Rcx	Rdx	rax	R8	rax

4. Dans quel cas a-t-on besoin de réaliser des temporaires avec des slots de pile ? Fixer le ‘ ?’ de ‘alloc ?’.

On a besoin de réaliser des temporaires avec des slots de pile quand il y a plus de temporaires que de registres.

Exercice 2 : Organisation des données structurées

Structure 1 : Il s'agit d'un tableau de 2 instances de la même structure.

```
struct {int a; int b;}[2] x;
```

x[0]
x[1]
x[0].a
x[0].b
x[1].a
x[1].b

Structure 2 :

```
struct {
     struct{int re; int im;} a;
     int[2] b;
}y;
```

y.a
y.b
y.a.re
y.a.im
y.b[0]
y.b[1]

Exercice 3 : Compilation des données structurées

```
typedef struct {int x; int y; } point_t;
typedef struct {point_t A; point_t B; } segment_t;

void init(segment_t s, int n)
{
     s.A.x = n;
     s.A.y = n;
     s.B.x = n+1;
     s.B.y = n+1;
     //P
}

void main(int argc, (char*)* argv)
{
     segment_t s;
     init(s,argc);
}
```

1. Donner l'état de la pile au point P

argc
argv
@retour
Old bp
s.A
s.B
s.A.x
s.A.y
s.B.x
s.B.y
Arg : s=s
Arg : n=argc
@retour
Old bp

2. Donner le prélude et le postlude de init

Prélude :
init ;
alloc ?
t1 = [bp+3]

t2 = [bp+2]
ρ(s)=t1, ρ(n)=t2
Postlude:
free
ret 2

3. Que fait le code $[[s.B\ .y]]_{ta,tv}$? Donner ce code et vérifier qu'il permet d'accéder à la donnée en question.

```
[[(s.B).y]]_ta,tv =
        [[s.B]]_tbase == [[(s).B]]_ta',tbase =
                [[s]]_tbase' =
                        tbase' = ρ(s) (=t1)
                toffset' = rang(B) (=1)
                ta' = tbase' – toffset'
                tbase = [ta']
        toffset = rang(y) (=1)
        ta = tbase – toffset
        tv = [ta]
```